23. (Amended) The method of Claim [21] 11, wherein tasks supplied with PCI models are mapped to extended Tcl functions, allowing execution of Tcl scripts that interact with other devices on a PCI bus.

24. (Amended) The method of Claim [21] 11, wherein Verilog code causes one of said Tcl interpreters to start executing a script when it senses an interrupt on a PCI bus.

25. (Amended) The method of Claim [21] 11, said method further comprising:

checking whether a PCI bus is busy before calling a task that starts a transaction on said bus; and

if said bus is busy, waiting for the current transaction to complete before starting a new transaction.

26. (Amended) The method of Claim [21] 11, said method further comprising:

providing a PCI_TCL module for extensive testing of any PCI based device without writing a single line of Verilog code for said test bench.

27. (Amended) The method of Claim [21] 1, said method further comprising:

providing a library of Tcl procedures that simplifies tasks.

28. (Amended) The method of Claim [21] 1, wherein functionality between Tcl and Verilog is partitioned such that said scripts are reusable in system level testing.

29. (Amended) The method of Claim [21] 1, said method further comprising:

providing a porting of Tcl scripts to real hardware; and

running a verification suite on ASICs when they return from a foundry.

36. (Amended) The apparatus of Claim 29, further comprising: